From mimikatz to kekeo

passing by new Microsoft Security Technologies

OPCDE_

by comae technologies

Benjamin DELPY `*gentilkiwi*`

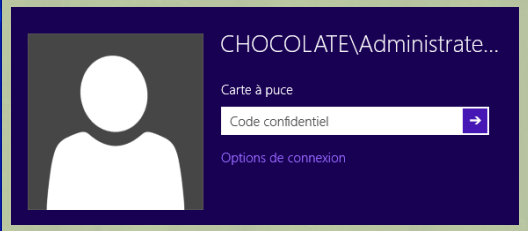
`whoami` ?

- **Benjamin DELPY - @gentilkiwi**

  - –Security researcher at night (*it's not my work*)
  - –Author of `mimikatz`
    - *This little program that I wrote to learn C*
    - And **kekeo**, for my (your ?) personal usage ;)
  - –I'm not:
    - **Bachelor, CISSP, CISA, OSCP, CHFI, CEH, ISO*, MCSA, CHFI, PASSI, […]**
  - –Memes count: **>15** – "Cyber*": **0**

mimikatz :: sekurlsa

LSA (PLAYSKOOL level)



WinLogon → LsaSS → Authentication

Authentication
- msv1_0 → SAM
- kerberos → LDAP

user:domain:password

Challenge / Response

Authentication Packages
- msv1_0
- tspkg
- wdigest
- livessp
- kerberos

Little retrospective

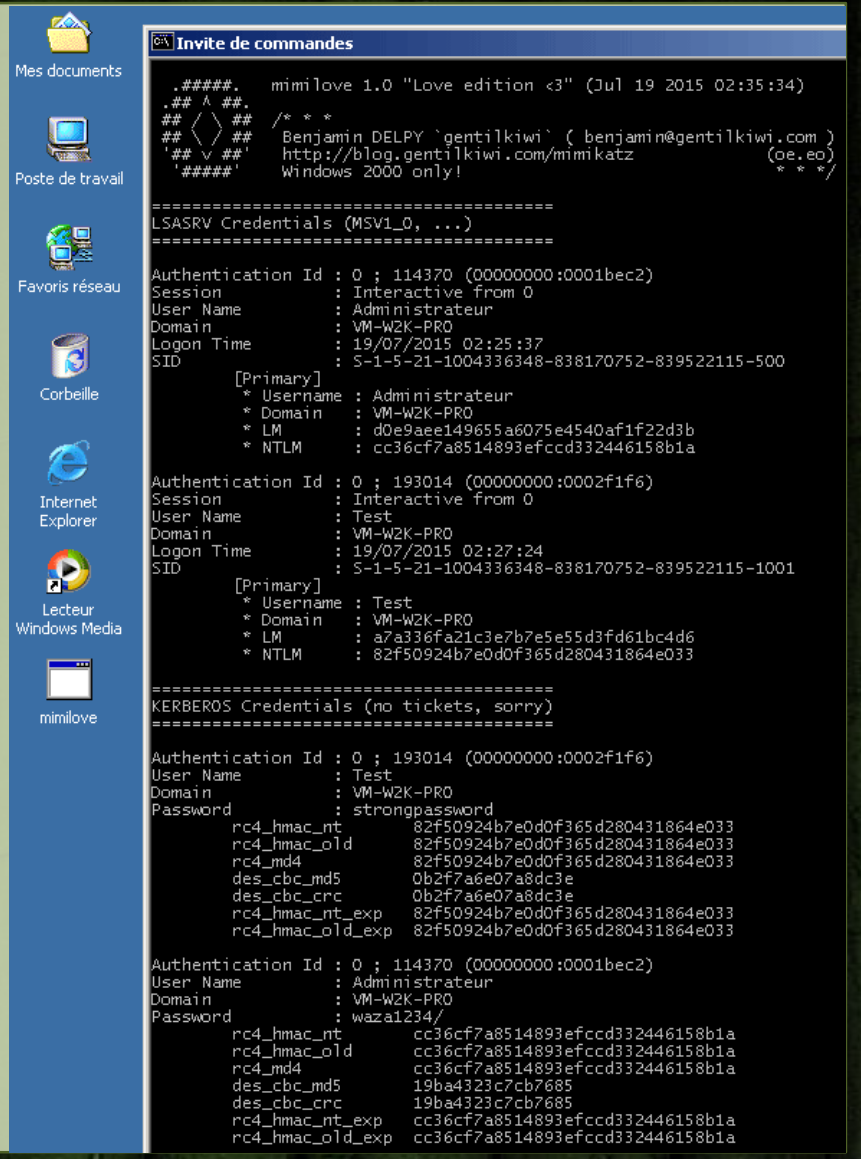
Windows 2000

- Yes, it still exist on some AD environments
  - Who said ‘even DC’ ? ☺
- Credentials can be in multiple places, but in LSASS we can easily find:
  - LM & NTLM hashes (no encryption at all) ;
  - Kerberos:
    - keys (no encryption at all) ;
    - Not tested: tickets and session key, but it seems logical ;
    - Passwords (if not already consumed) ;
- Passwords are “encrypted” in memory

```
hash = *((PBYTE) &session.Password.Length + 1);
*((PBYTE) &session.Password.Length + 1) = 0;
RtlRunDecodeUnicodeString(hash, &session.Password);
```

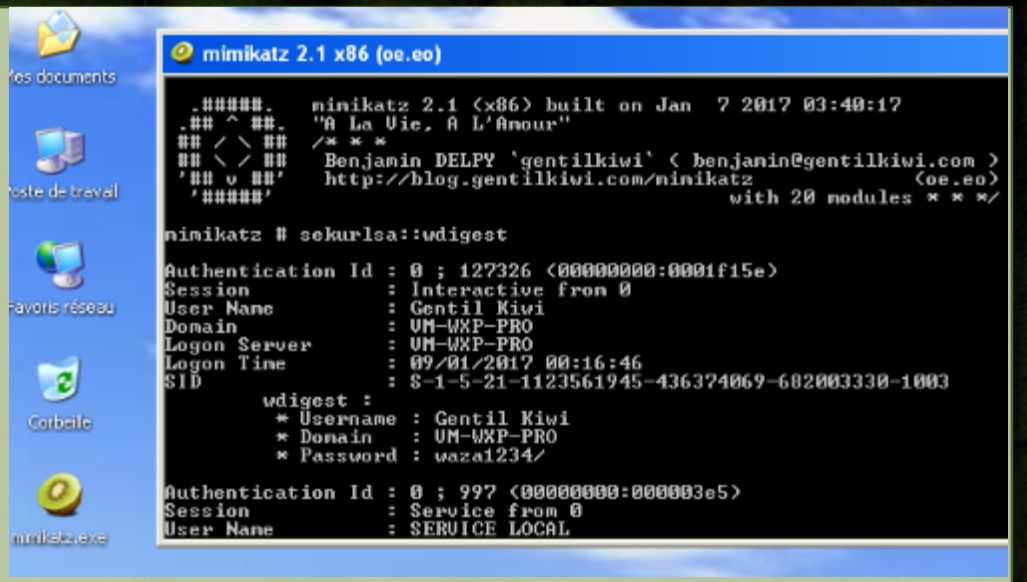
basically, a XOR with a 1 byte ‘key’ !

- *And this “key” is stored in the secret structure*



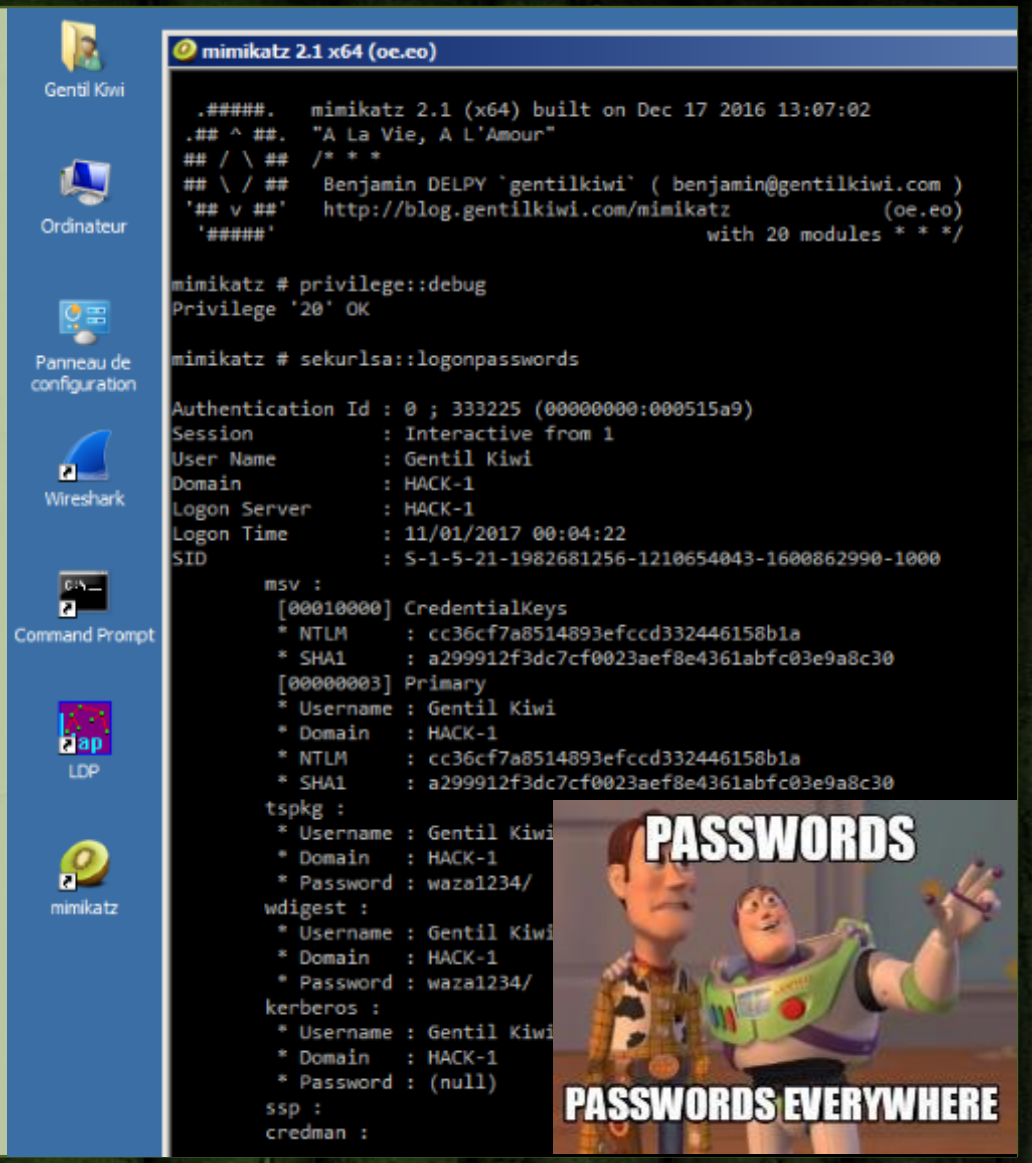
Windows XP/2003

- This time, we have a gift: **WDigest** provider
  - *Authentication to Web Site, SASL, LDAP scenarios…*
  - https://technet.microsoft.com/library/cc778868.aspx
  - https://www.ietf.org/rfc/rfc2617.txt
- We now have constantly passwords in memory
  - Someone at Microsoft had to think it was dangerous
- LSA SSO secrets are now protected by **LsaEncryptMemory**
  - and unprotected by **LsaUnprotectMemory**
  - Crypto: **RC4** or **DESx** algorithms (depending on the secret size)
- *Keys and 'IV' are stored near the secret (in LSASS process)*
- *First Windows version supported by mimikatz to dump Kerberos tickets and associated session key*
- In "bonus", **TsPkg** provider can be added manually in Windows XP (but let's go on)



Windows Vista/7

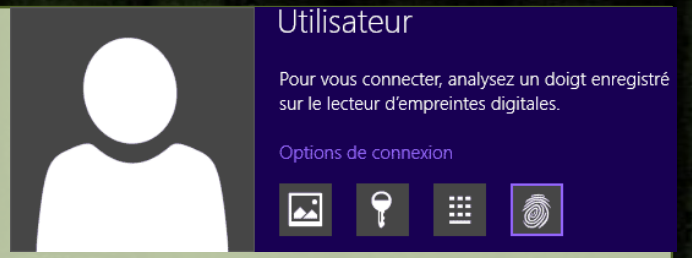
- Other new gifts
  - Older still here, of course ;)
  - ***TsPkg*** *(CredSSP support)*
    - *Credential Delegation for Terminal Server/PowerShell/Double hop, etc…*
  - ***LiveSSP*** *(ok, not really a gift, but it's here too)*
- We now have constantly several passwords in memory
  - Someone at Microsoft had to think it was more and more dangerous
- LSA SSO secrets are still protected by **`LsaEncryptMemory`**
  - and of course unprotected by **`LsaUnprotectMemory`**
  - Crypto: **3DES** or **AES** algorithms (depending on the secret size)
- *Keys and IV are stored near the secret (in LSASS process)*



Windows 8/8.1

“Sometimes it is necessary to hit the rock bottom to better bounce”

- Still no change… a lots of passwords, keys, tickets



- Some new crazy stuff at logon:
  - PIN Code
  - Picture
    - guys, really?
  - Fingerprints

Domain credentials in CLEAR in system vault !

Offline access is possible

« lol »

- Default “(over)pass-the-hash” or “pass-the-ticket” to Remote Desktop

Windows 8/8.1

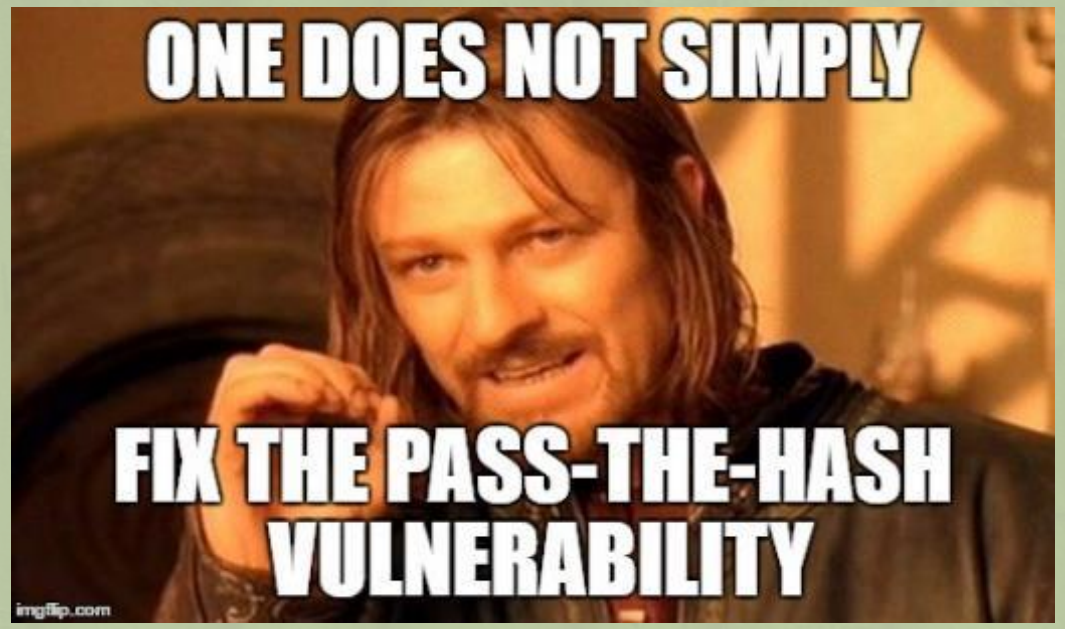
“Sometimes it is necessary to hit the rock bottom to better bounce”

- Windows 8.1 introduced a new “clean base”
  - **No password by default in memory**
    - WDigest is now “off” by default ☺
  - **LSA logon session cache cleaner**
    - it’s more difficult to get credentials after logoff
  - **“Restricted Admin mode for Remote Desktop Connection”**
    - Avoid user credentials to be sent to the server (and stolen)
    - Allow authentication by **pass-the-hash**, **pass-the-ticket** & **overpass-the-hash** with **CredSSP**
  - **“LSA Protection”**
    - Deny memory access to **LSASS** process (protected process)
    - Bypassed by a driver or another protected process (remember? **mimikatz** has a driver ;))
  - **“Protected Users security group”**
    - No more **NTLM**, **WDigest**, **CredSSP**, no delegation nor SSO... Strengthening **Kerberos** only!
    - Kerberos tickets can still be stolen and replayed (and smartcard/pin code is in memory =))
- KB 2975625
    - **Restricted Admin is now disabled by default ;)**
- Bad stuff still here for compatibility ☹

Windows 8/8.1

“Sometimes it is necessary to hit the rock bottom to better bounce”

- But management and marketing was here
  - *Management is always here… (Tal too)*



Before:

Microsoft Support

Account Sign in

Microsoft Security Advisory: Update to fix the Pass-The-Hash Vulnerability: May 13, 2014

Print Email

Article ID: 2871997 View products that this article applies to.

After:

Microsoft Support

Account Sign in

By product Downloads Store Contact us

Microsoft Security Advisory: Update to improve credentials protection and management: May 13, 2014

Print Email

Article translations

Article ID: 2871997 View products that this article applies to.

Tal Be'ery @TalBeerySec · 18 mai 2014

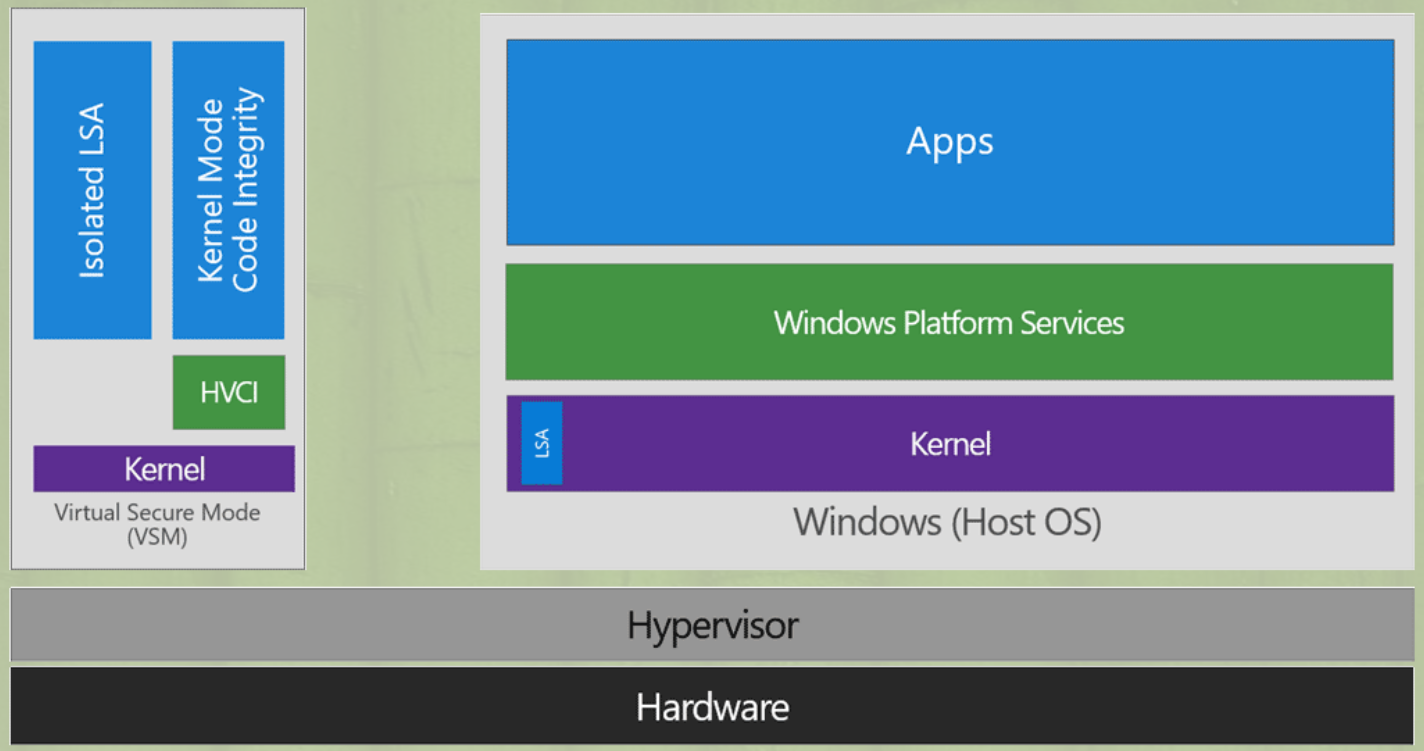
Can you spot the differences? #Microsoft #KB2871997. #PtH is alive! webcache.googleusercontent.com/search?q=cache..., support.microsoft.com/kb/2871997

5 50 16 •••

Windows 10

“A new hope”

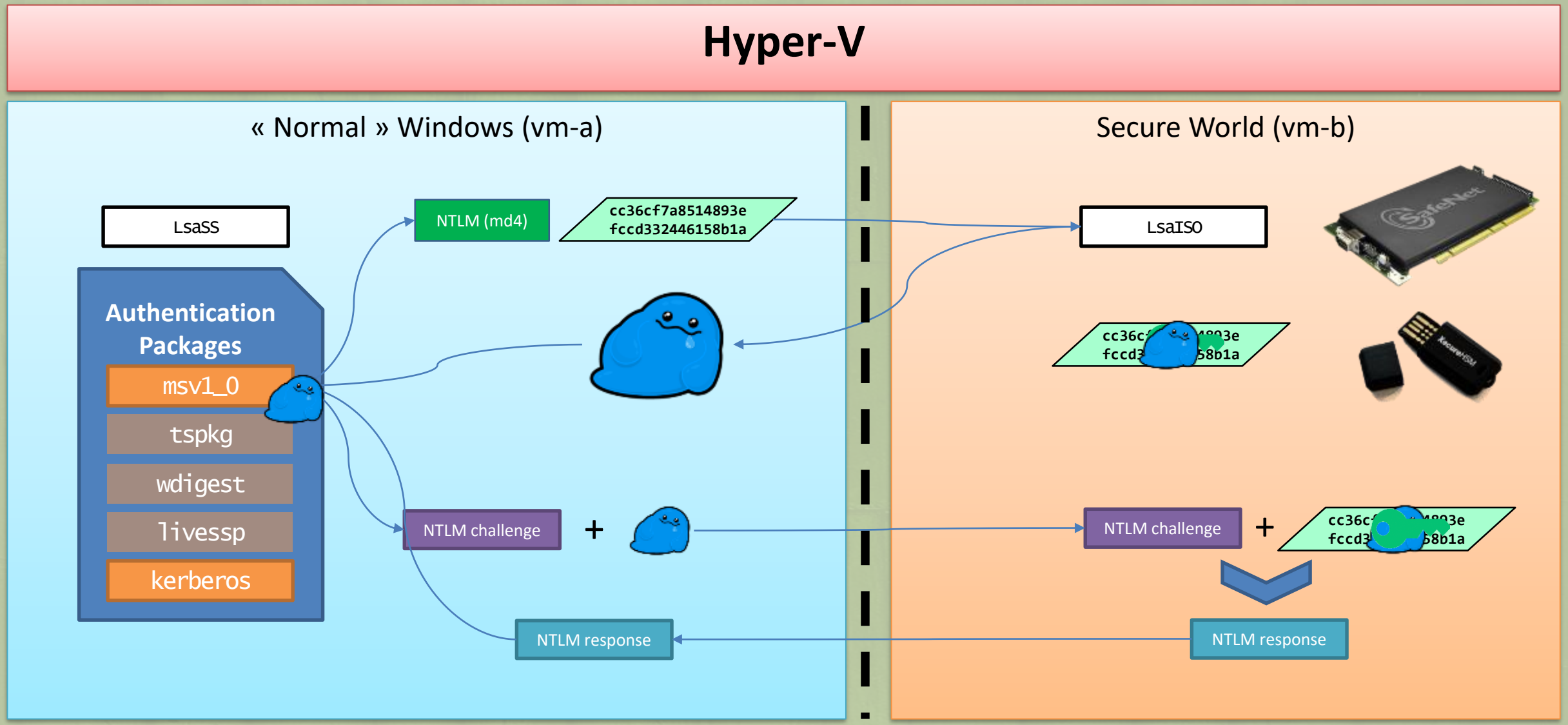
- We keep all the good things and we will add a secret ingredient: VSM
  - *But only for the rich, for the others: telemetry, Cortana…*
- A very good move *approaching* Crypto HSM architecture !



https://blogs.technet.microsoft.com/ash/2016/03/02/windows-10-device-guard-and-credential-guard-demystified/

Windows 10 - Credential Guard

NTLM - *without B.S.*



Windows 10 - Credential Guard
Kerberos - *without B.S.*

Hyper-V

« Normal » Windows (vm-a)

LsaSS

Authentication Packages

- msv1_0
- tspkg
- wdigest
- livessp
- kerberos

Authentication Key
(password derived)

rc4_hmac_nt (NTLM/md4)	cc36cf7a8514893e fccd332446158b1a
aes128_hmac	8451bb37aa6d7ce3 d2a5c2d24d317af3
aes256_hmac	1a7ddce7264573ae 1f498ff41614cc78 001cbf6e3142857c ce2566ce74a7f25b

TGT Session Key

TGT

Secure World (vm-b)

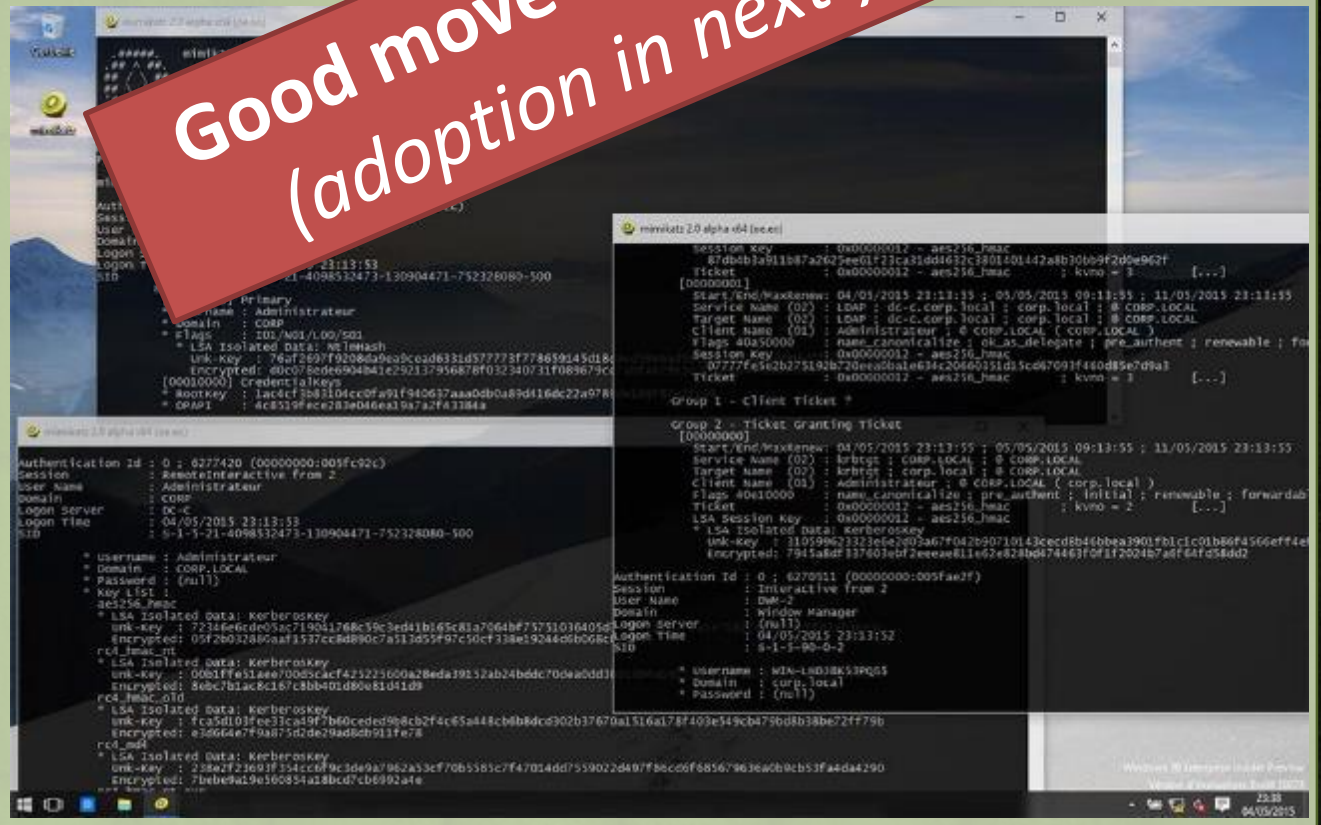
LsaISO

Windows 10 - Credential Guard
without B.S.

- **Prerequisites** (at this time)
  - Windows 10 Enterprise
  - ~~Domain (works only with domain accounts)~~
  - SecureBoot
  - Virtualization
    - Don’t start VMWare in the « guest » Windows OS after ;)
  - ~~*TPM 2*~~

- **Limitations**
  - ~~No local accounts~~
  - No Kerberos TGS session keys (why ?)
  - Not really available in Virtual Machine (for now)
  - Not enable by default
  - Rely on Hyper-V and Hardware protection *(Intel © love to flash bad firmware ;))*

Good move from Microsoft !
(adoption in next years?)



- **A lot of potential**
  - Remote Credential Guard (cool ☺)
    - Kerberos
    - NTLM ?
  - SmartCard PIN code ?
  - DPAPI ? (no, not the NG)
    - Private Key (even legacy ones)
    - Domain Passwords ?
  - SAM ?
    - Yep, even DSRM,
  - Domain controller sensitive operations ?
    - Kerberos PAC Signature ?

What now ?

- Previous credentials stealing techniques can’t work anymore in the same way
  - I’m lazy (*attackers too*)
- There are several ways to obtain information or gain privileges by normal protocols/operations and logical approach
  - **[MS-DRSR]** - Directory Replication Service
  - **[MS-NRPC]** - Netlogon Remote Protocol
  - **[MS-BKRP]** - BackupKey Remote Protocol
  - **[MS-LSAD]** - Local Security Authority (Domain Policy) Remote Protocol
  - **[MS-PAC]** - Privilege Attribute Certificate Data Structure
  - Kerberos
    - **[MS-KILE]** - Kerberos Protocol Extensions
    - **[MS-SFU]** - Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol
  - …
- By example: Duqu 2.0 @ Kaspersky relied on **a very logical attack** on Microsoft PAC Signature in Kerberos tickets (MS14-068 – user to Domain Admin/Enterprise Admin)
  - https://securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf

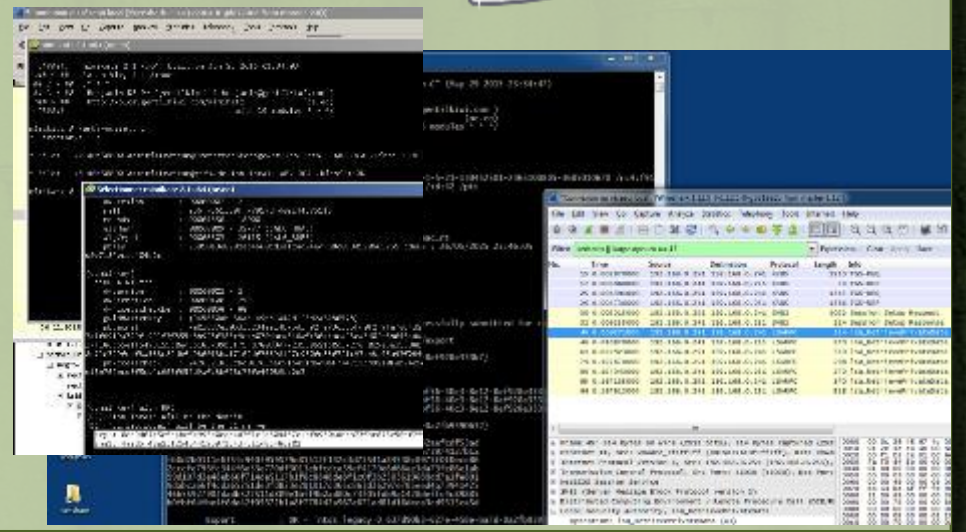
A lots of sexy names!

Various methods

- Golden tickets
  - Access to all the domain with 1 secret ☺
  - *Who said « forest »?*

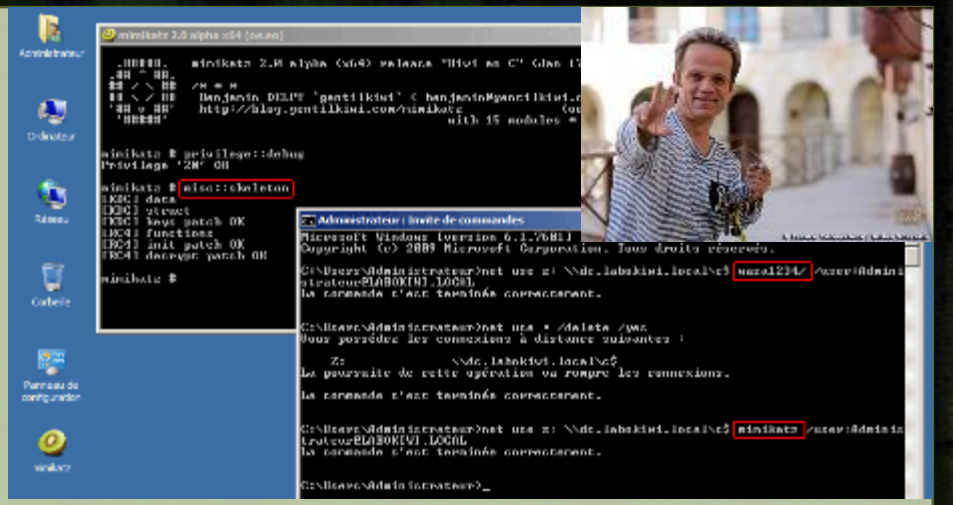
- Silver tickets
  - Access to a service (usualy a server/computer)
  - Silver is the new ~~black~~ pass-the-hash

- DPAPI Backup key:
  - Offline: by retrieving the "***~never changed master DPAPI key***"
    - Access to secrets of all users of the domains
  - Online: accessing our own secrets without knowing our password
    - Domain Passwords too
    - Useful for malware ☺



Various methods

- Skeleton Key: why steel passwords when you can use your own everywhere?
  - https://www.secureworks.com/research/skeleton-key-malware-analysis



- Play with your SID !
  - With sidHistory : be “Domain admins” without being “Domain admins”
  - Transform yourself as a domain controller to avoid logs ☺

mimikatz :: DCSync

- **DCSync** was originally a standalone program co-developed with **Vincent LE TOUX** (http://www.mysmartlogon.com)
- It was followed by
  - Impacket, **Alberto Solino** (https://github.com/CoreSecurity/impacket)
  - DSInternals, **Michael Grafnetter** (https://www.dsinternals.com/en/)
    - **These guys rock !**
- **DCSync** is now fully integrated in **mimikatz** (*lsadump::dcsync*) to take benefit of all its functions

https://twitter.com/subTee/status/637469004987129856

mimikatz :: DCSync

- **DCSync** is a new way to get Credential from the Active Directory…
  - Without file on disk or memory
  - Without shellcode in memory
  - Without process dumping
  - Without filebackup/shadow copy/physical access.

- Like others, it needs some specific rights
  - Administrator
  - Domain controller

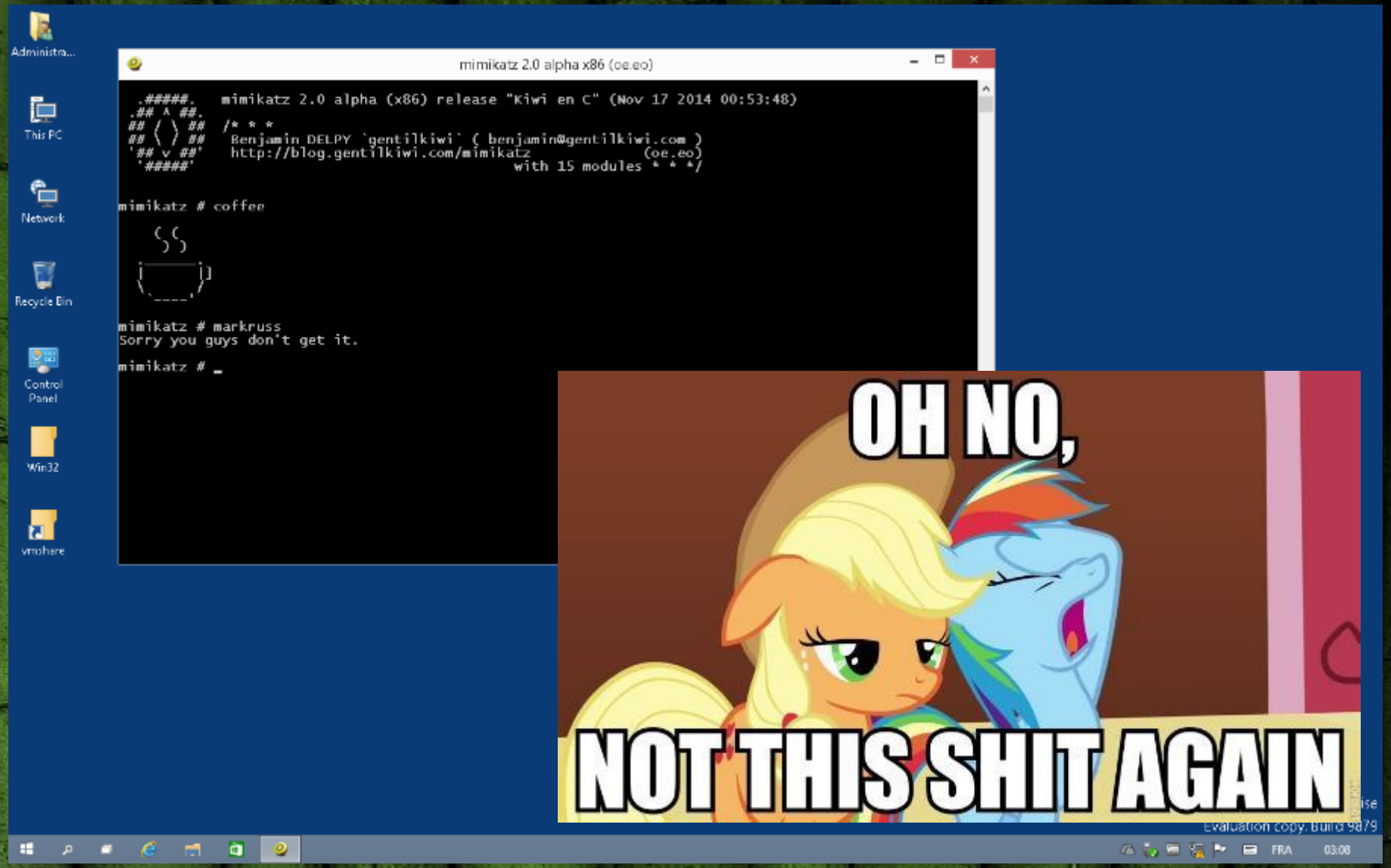
- **All is made remotely, by official RPC**
  - **default: no LOG when using a DC account !**

mimikatz :: DCSync

- **DCSync** will “mimic” API calls used in DC synchronization (**MS-DRSR**).
  - DCs exchange objects modification. One modification can be, of course, sensitive data

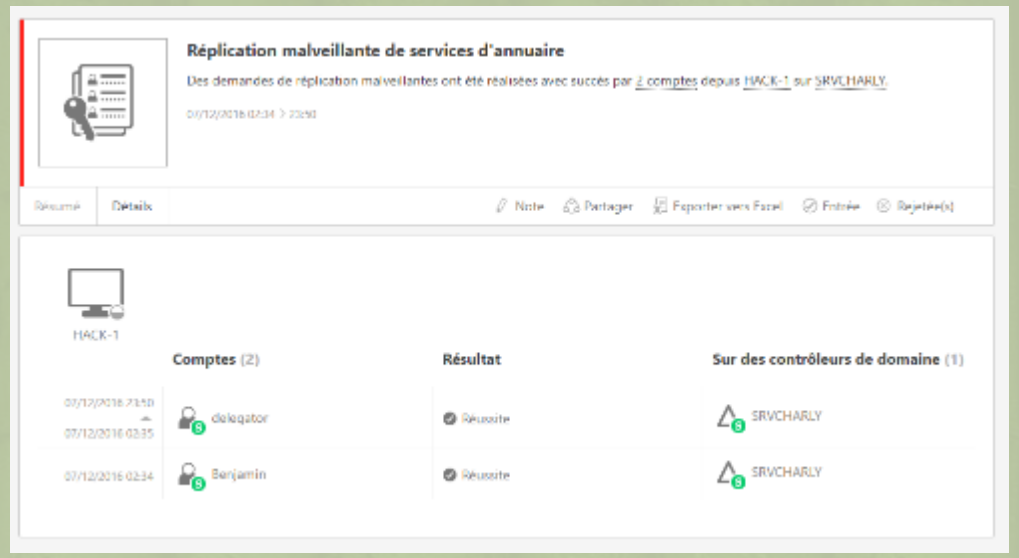
DRSBind (standard)	• To get a handle to the DC as a normal user (it’s enough here)
DRSDomainControllerInfo	• To get NtdsDsaObjectGuid of one DC in the domain
DRSCrackNames	• To translate the user/object name in a GUID
DRSUnbind (standard)	• To release the standard handle (we don’t need it anymore)
DRSBind (DC)	• To get a handle to the DC as another DC (using NtdsDsaObjectGuid)
DRSGetNCChanges	• To obtain all change of the targeted object (using its GUID)
DRSUnbind (DC)	• To release the DC handle (we don’t need it anymore)

Demo !

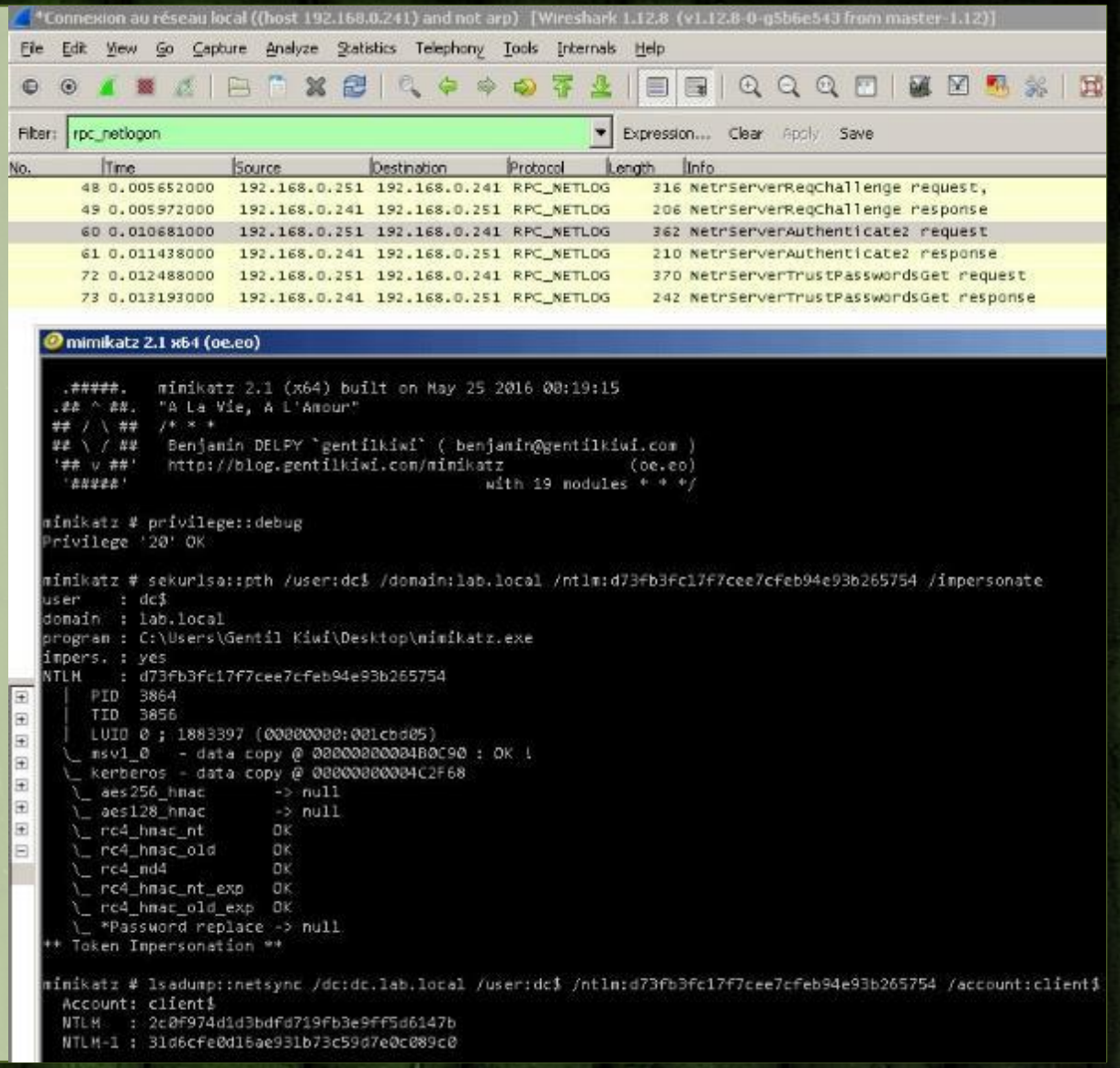

mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz # _
OH NO,
NOT THIS SHIT AGAIN

mimikatz :: netsync

- **[MS-DRSR]** DCSync is easy to spot:



- **[MS-NRPC]** What about NetLogon ? ☺
  - ***Now*** limited to 'not user accounts', but:
    - Silver tickets are so cool
    - Account type can be modified, even temporary



kekeo

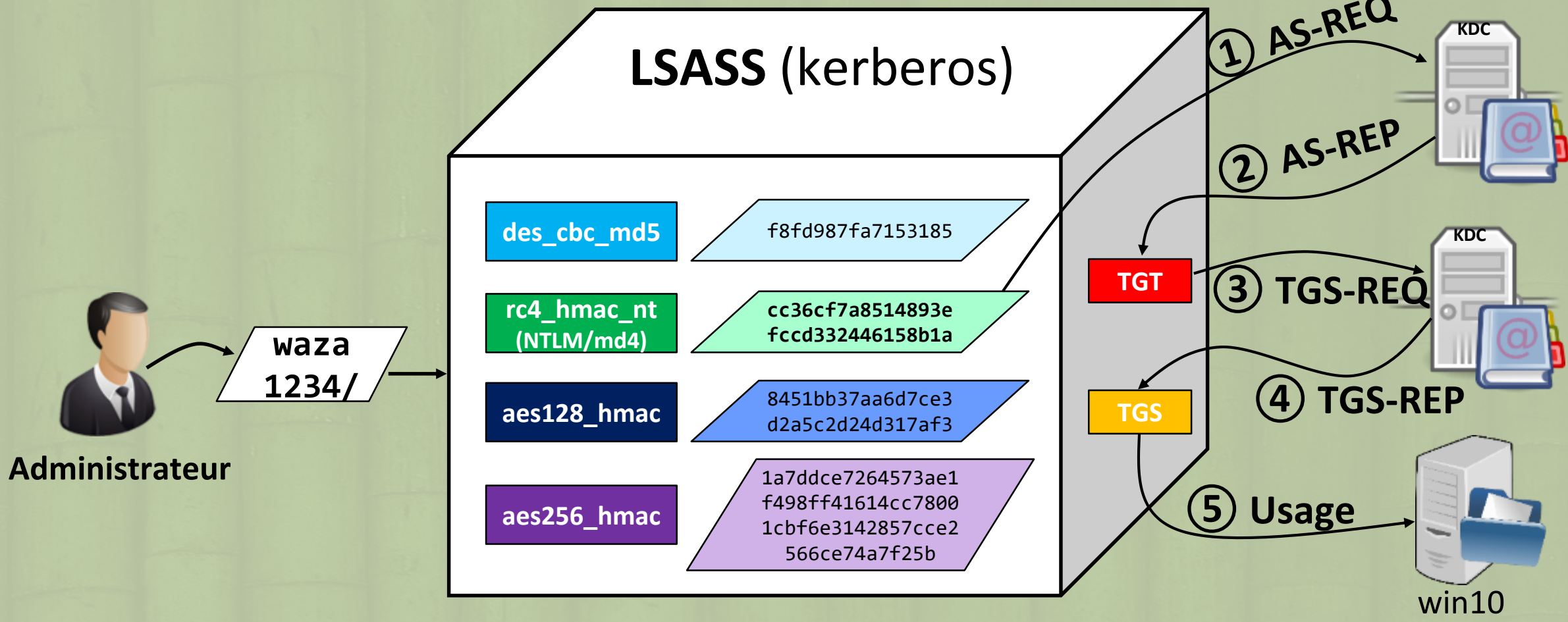
- **kekeo** is a Kerberos tools suite ☺

- I've made it external to **mimikatz** for two reasons
  - I hate to code network related stuff ;
  - It uses an external commercial ASN.1 library inside
    - It was hard for me to make it works with others…but **OSS/Nokalva** teams was kind enough to offer me a 'limited' license ☺
    - It seems quite close to some Microsoft code ;)

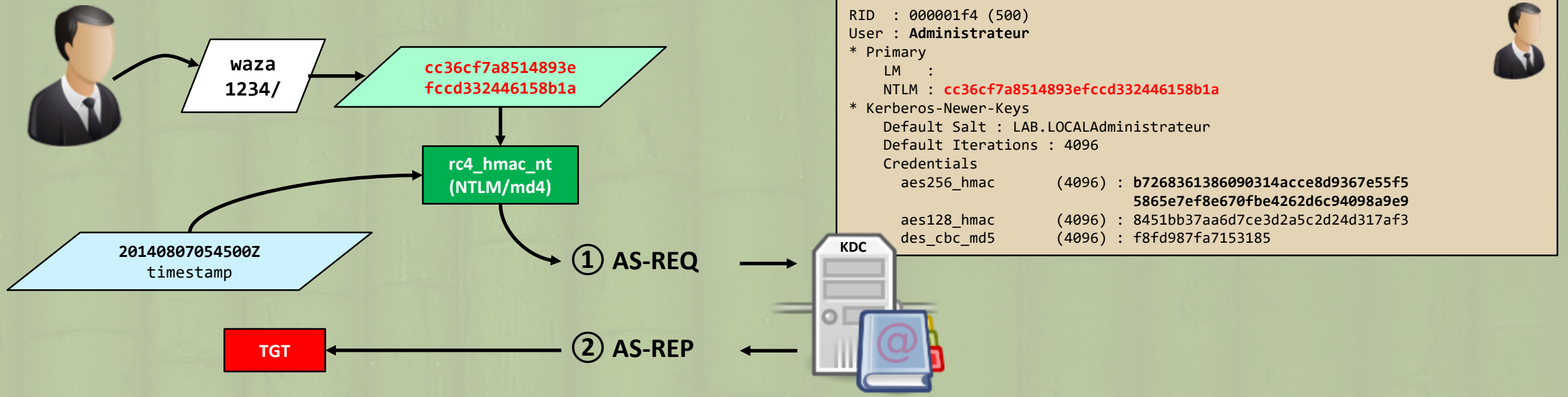
- It includes
  - **ms14068** to exploit MS14-068 (authenticated user to DA/EA/DC$ avoiding KDC signature check)
  - **ms11013** to exploit MS11-013 (authenticated user to DA/EA/DC$ avoiding * signature check)
  - **asktgt** to authenticate against a KDC with various methods (*including* ***pkinitmustiness*** *stuff*), and get a TGT
  - **asktgs** to request service tickets from a TGT (or renew)
  - **aoratopw** to change a password for a user from its keberos key (not reseting it)
  - **kirbikator** to convert tickets between kirbi/wce/ccache/LSA format
  - **pkinitmustiness** ☺
  - **s4u** to play with delegation (S4U2Self & S4U2Proxy)
  - **kerbstorm** /

A little reminder about Kerberos authentication

- How does it works ?



A little reminder about Kerberos authentication



- The **KDC** will validate the authentication if it can decrypt the timestamp with the long-term user key (for **RC4**, the **NTLM** hash of the user password)
- It issues a **TGT** representing the user in the domain, for a specified period

A little reminder about Kerberos authentication

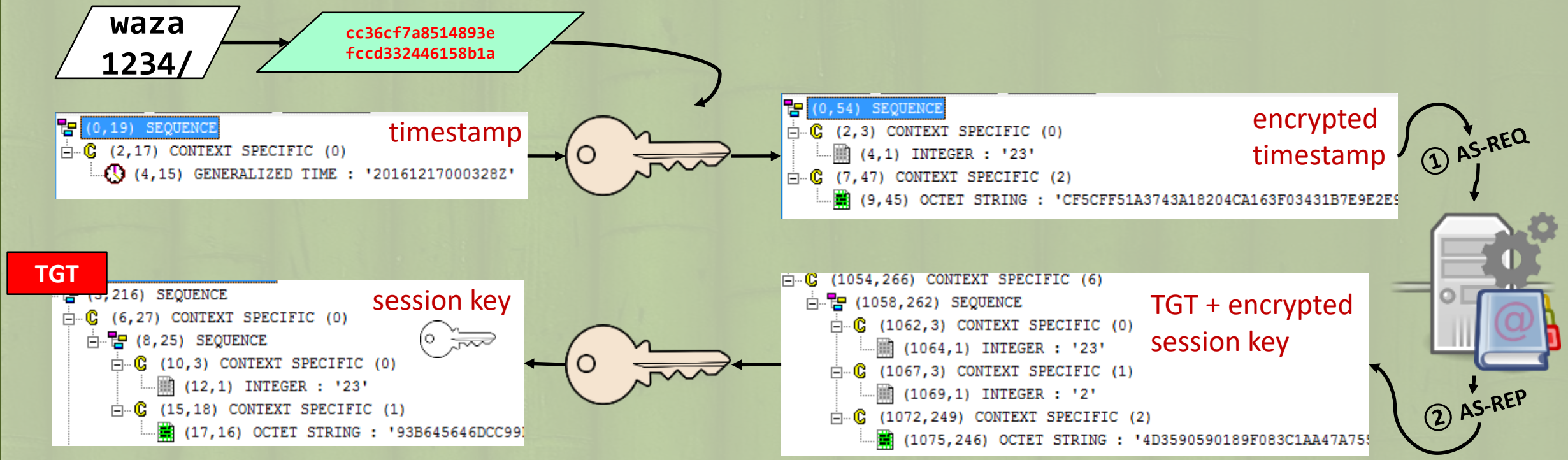
- But what about Smartcard, Token, …

Kerberos authentication

Kerberos authentication

Password mode

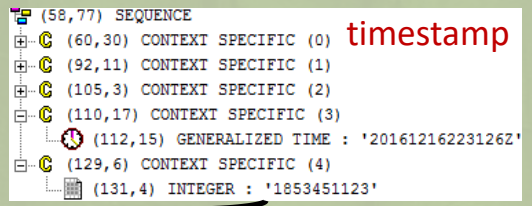
- Passwords lead to symmetric keys

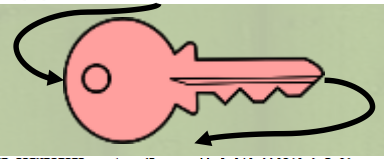


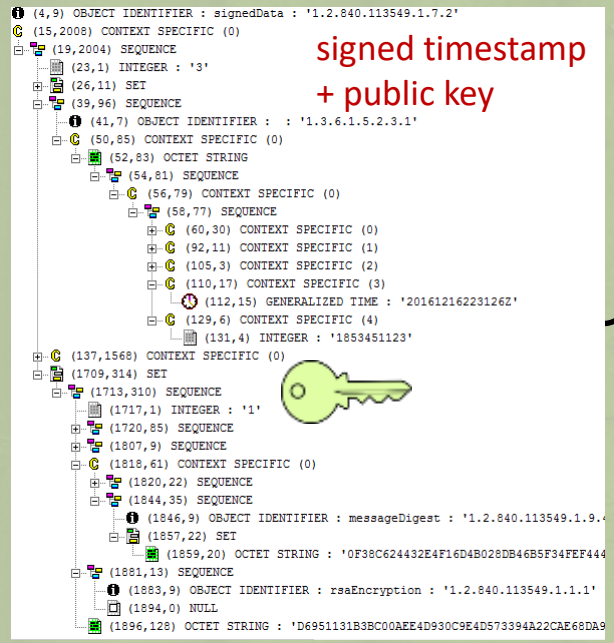
Kerberos authentication

RSA Mode

- Smartcards/tokens lead to asymmetric keys





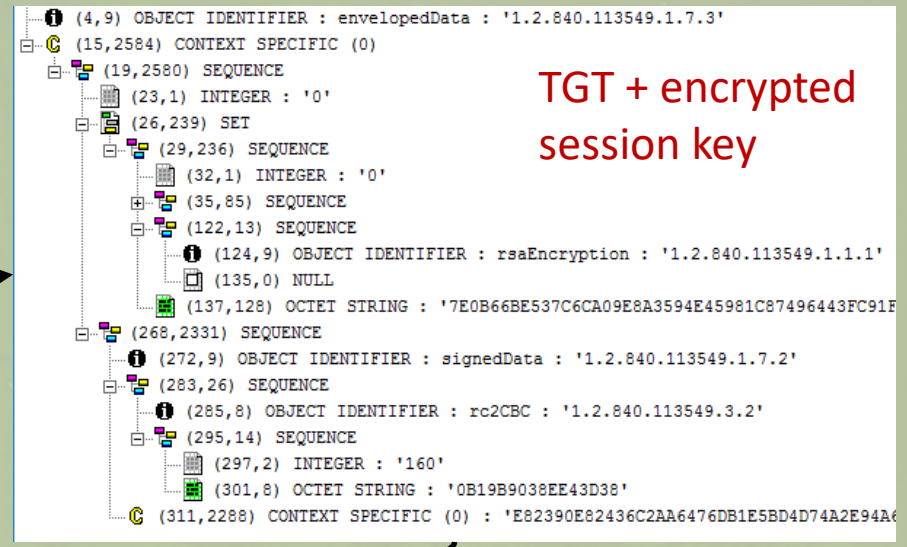


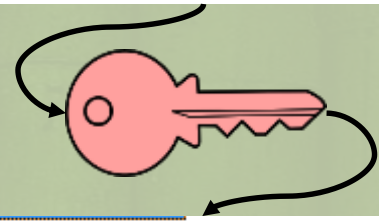
Private Key

Public Key

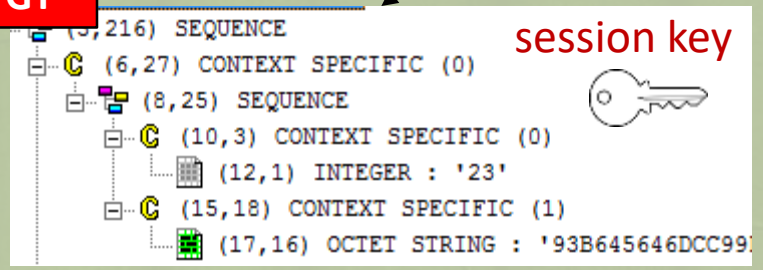
① AS-REQ

② AS-REP





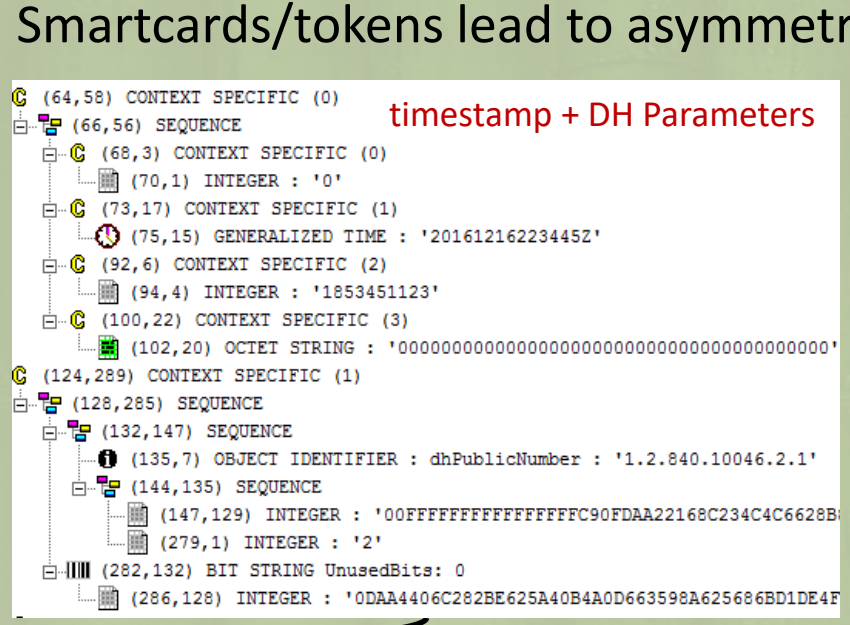
TGT



Kerberos authentication

*RSA Mode + **Diffie–Hellman***

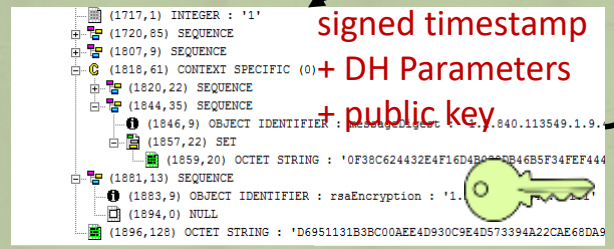
- Smartcards/tokens lead to asymmetric keys



timestamp + DH Parameters

Private Key

Public Key

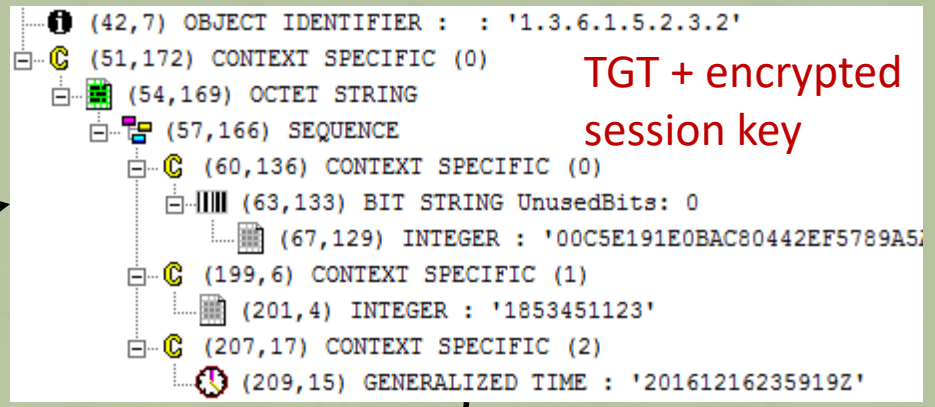


signed timestamp
+ DH Parameters
+ public key

① AS-REQ

Diffie-Hellman

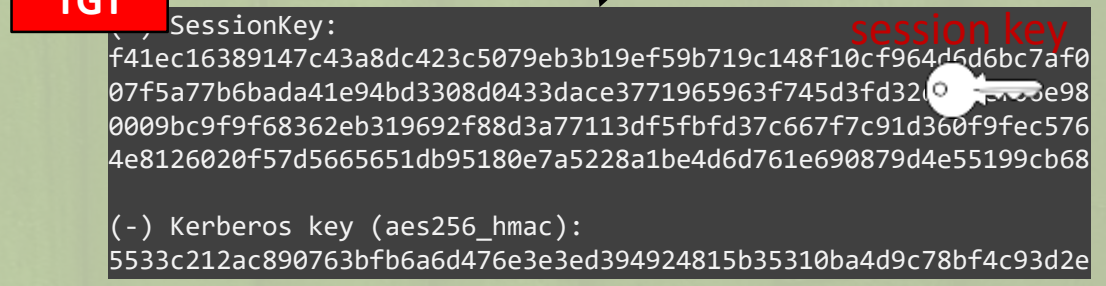
② AS-REP



TGT + encrypted session key

Diffie-Hellman

TGT



session key

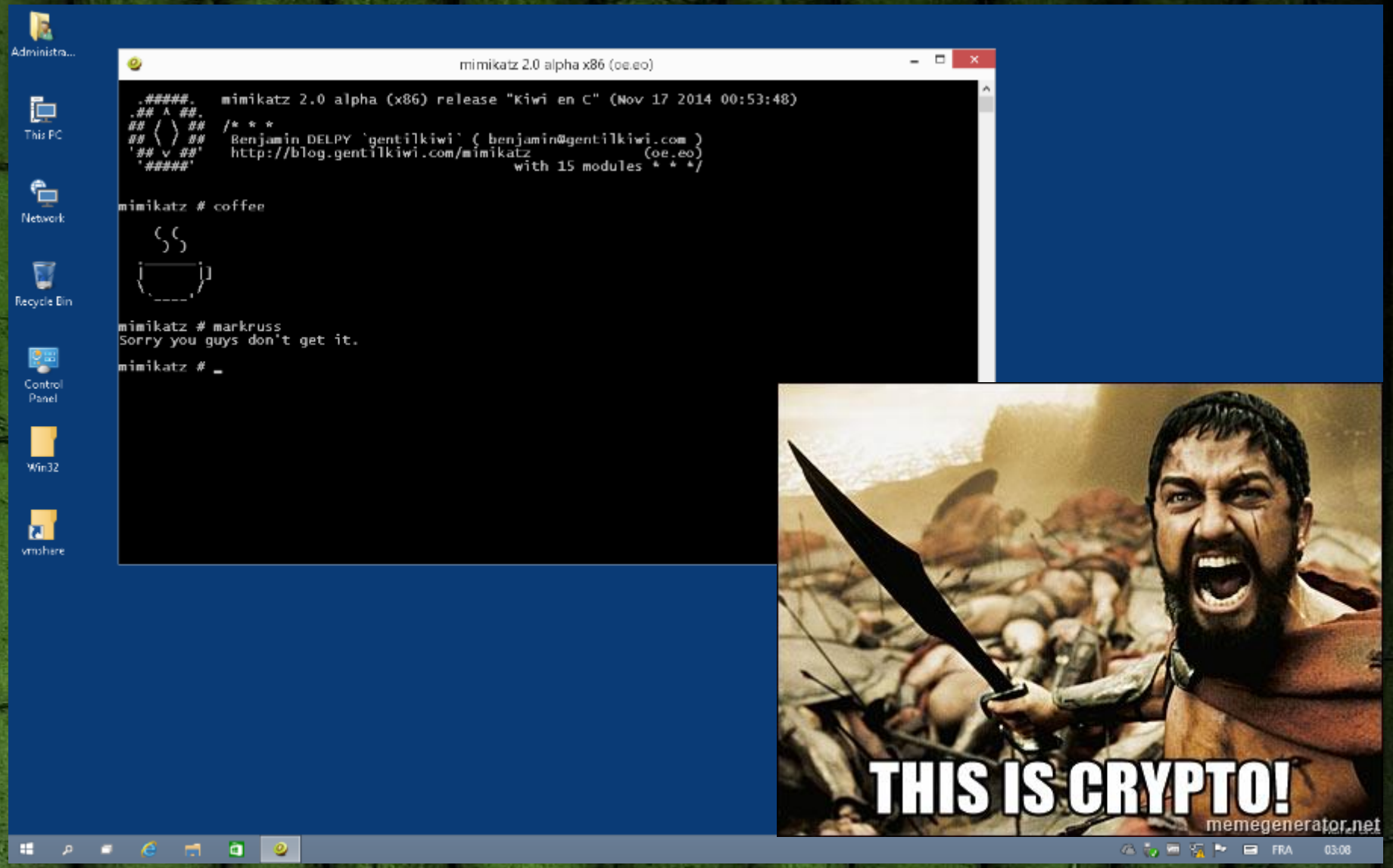
Kerberos authentication

Mode	Secret needed to encode AS-REQ	Secret needed to decode AS-REP
Password / Key	YES	YES
RSA	YES	YES
RSA with Diffie-Hellman	YES	**NO**

- **Once we have access to the Smartcard/Token, even for a short time, we can generate multiple pre-signed AS-REQ for future usage ☺**
  - as long as the source certificate validity (usually seen « years »)

- Do you remember ? Windows LSA service **keeps PIN code in memory**
  - Useful on Terminal Server where LSASS can control remote Smartcards (even virtual ones) ;)

Demo !


mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
THIS IS CRYPTO!
memegenerator.net

Kerberos authentication

RSA Mode + Diffie–Hellman

- Is this Windows specific : **NO**
  - RFC 4556 :

```
3.1.1.  Required Algorithms

   All PKINIT implementations MUST support the following algorithms:

   o  AS reply key enctypes: aes128-cts-hmac-sha1-96 and aes256-cts-
      hmac-sha1-96 [RFC3962].

   o  Signature algorithm: sha-1WithRSAEncryption [RFC3370].

   o  AS reply key delivery method: the Diffie-Hellman key delivery
      method, as described in Section 3.2.3.1.
```

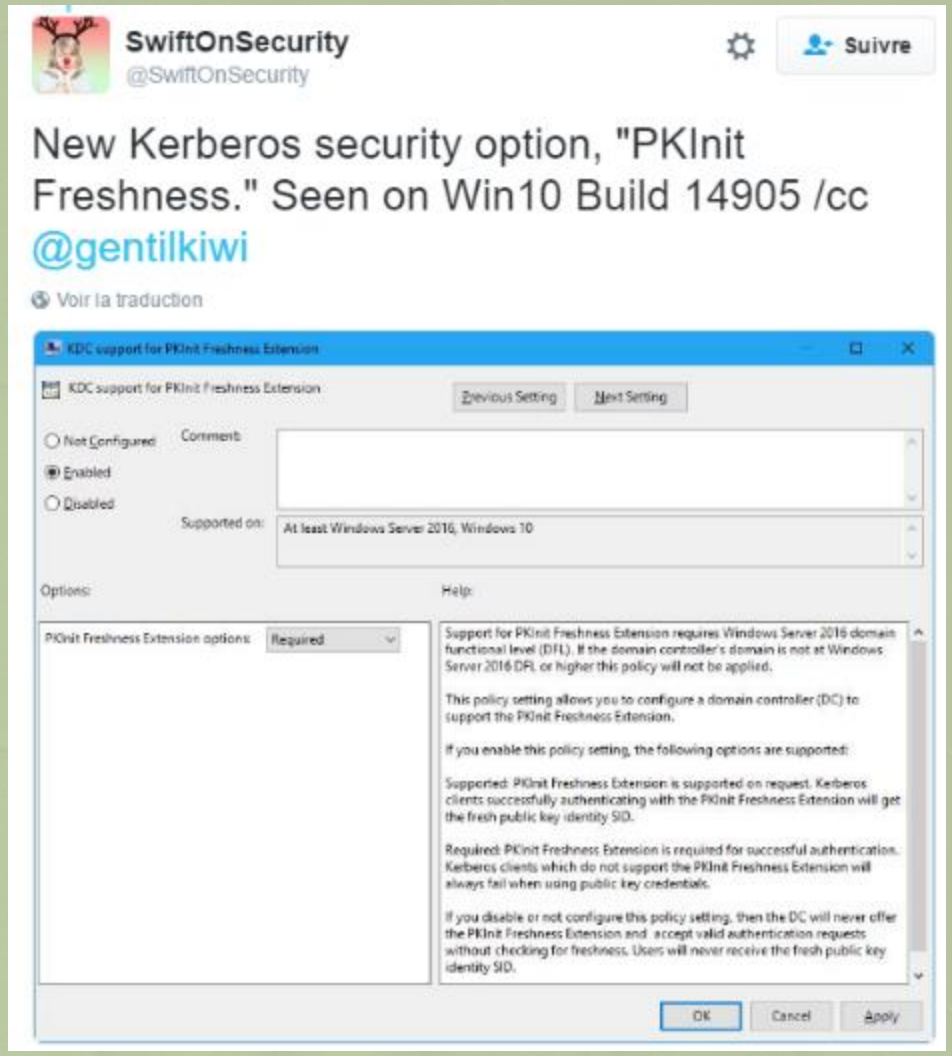
  - RFC 5349

```
   This document describes the use of Elliptic Curve certificates,
   Elliptic Curve signature schemes and Elliptic Curve Diffie-Hellman
   (ECDH) key agreement within the framework of PKINIT
```

Kerberos authentication

RSA Mode + Diffie–Hellman

- And what we can do?
  - Microsoft try to improve current Kerberos protocol by RFC draft:
    - https://datatracker.ietf.org/doc/draft-ietf-kitten-pkinit-freshness/
    - https://www.ietf.org/proceedings/91/slides/slides-91-kitten-1.pdf
  - They already implemented GPO for that (not tested) :
    - **But you must have a full network aware of it… (>= Windows 10 & 2016)**
  - Unless you use ECC certificates, it's not common to use DH with RSA certificates in Windows environment
    - Push some IPS rules to inspect AS-REQ… it's signed, NOT encrypted!
    - Windows Event Log does not seems to make differences between RSA and RSA/DH ☹



Kekeo bonus question?

- Why do I use in my code:

```
if(CryptAcquireContext(&keyInfo->hProv, NULL, MS_ENH_DSS_DH_PROV, PROV_DSS_DH, CRYPT_VERIFYCONTEXT))
{
        ((PDWORD) (((PULONG_PTR) keyInfo->hProv)[28] ^ 0xa2491d83))[5] = 1; // :)
        ...
```

kekeo – Cert to NTLM ?

- By default, when using SmartCard logon, NTLM hash is in memory for retro-comp. stuff
  - *not when using protected users*

- How does it works ?

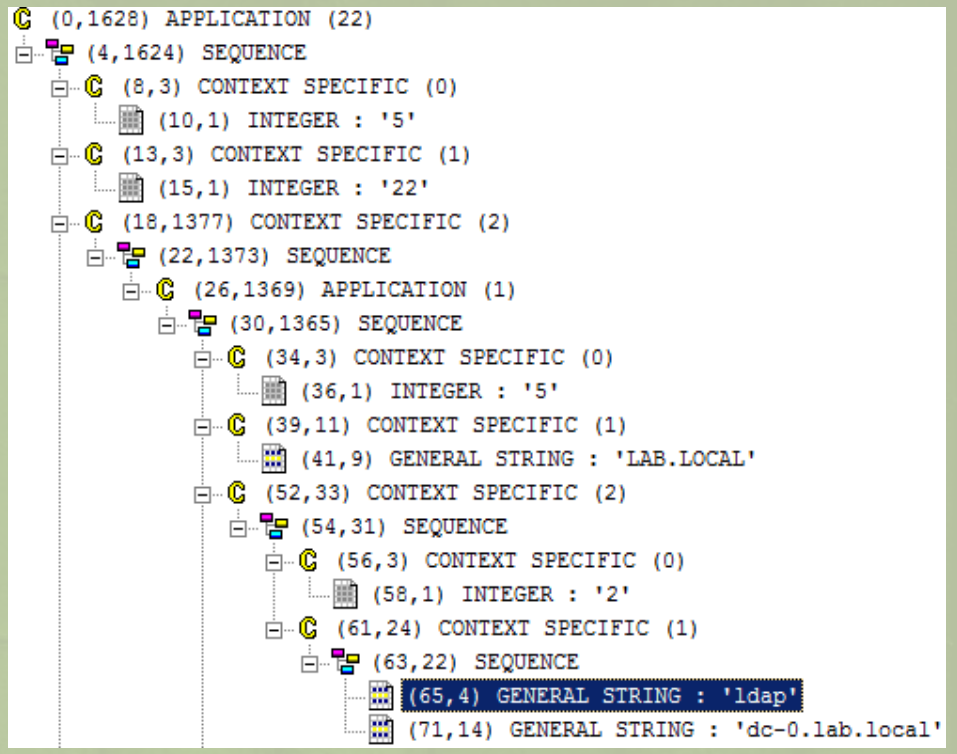
- **[MS-PKCA] to the rescue !**

> In order to support NTLM authentication [MS-NLMP] for applications connecting to network **services** that do not support Kerberos authentication, when PKCA is used, the KDC returns the user's NTLM **one-way function (OWF)** in the **privilege attribute certificate (PAC)** PAC_CREDENTIAL_INFO buffer ([MS-PAC] section 2.6.1).

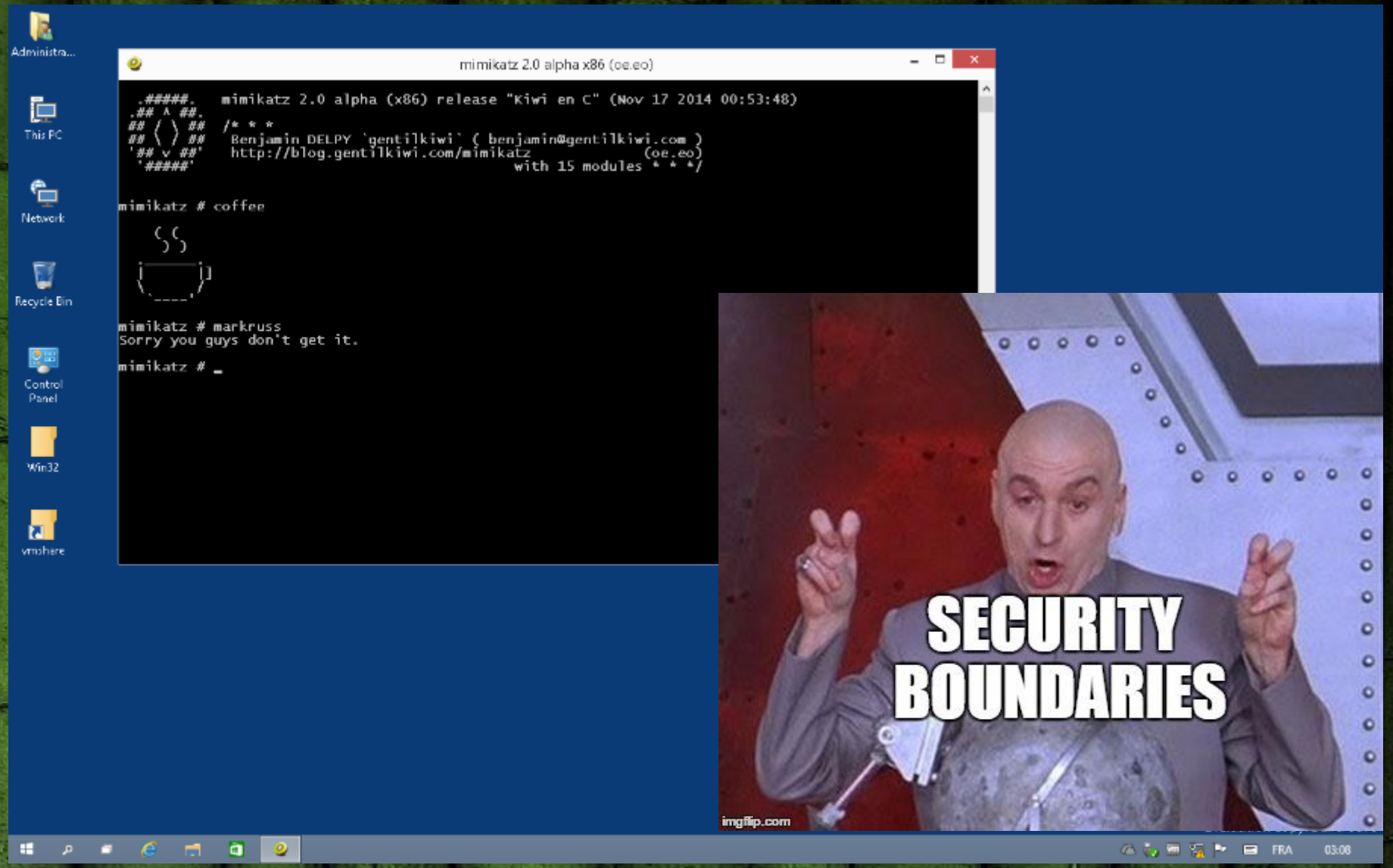
```
mimikatz 2.1.1 x86 (oe.eo)
User Name         : admin
Domain            : LAB
Logon Server      : DC-0
Logon Time        : 27/04/2017 09:24:53
SID               : S-1-5-21-412031729-2859336904-20738809
        msv :
         [00000003] Primary
         * Username : admin
         * Domain   : LAB
         * NTLM     : c3335430a69f1ec3f8bfa2314899bd5b
         * DPAPI    : 2137eb2ea31ce320be8117e4f73dcbbc
        tspkg :
        wdigest :
         * Username : admin
         * Domain   : LAB
         * Password : (null)
        kerberos :
         * Username : admin
         * Domain   : LAB.LOCAL
         * Password : (null)
         * Smartcard
             PIN code : 0000
             Card     : IDPrime MD T=0
             Reader   : OMNIKEY CardMan 3x21 0
             Container: te-KiwiCaP-2024b5fb-1ef1-46e1-a73-
             Provider : Microsoft Base Smart Card Crypto P
```

kekeo – “ServiceName is not a boundary”

- Thanks to Alberto Solino (CoreSecurity) we now know that the SPN in a ticket is not a security boundary.
  - *The impacket’s guy*
  - https://www.coresecurity.com/blog/kerberos-delegation-spns-and-more
- In fact, only the key used to encrypt ticket is the boundary
  - Yes, all services using the same key can share the same ticket…
- In impacket, Alberto can force its function to explicitly use a particular ticket.
  - I can’t (easily) make the same in Windows
- And what ?



Demo !


mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
SECURITY BOUNDARIES
imgflip.com

That's all Folks!

شكرا

- mimikatz http://blog.gentilkiwi.com/mimikatz (French)
- source https://github.com/gentilkiwi ← go here for **English** Wiki & Release (kekeo too) ;)
- contact @gentilkiwi / benjamin@gentilkiwi.com